AUTENTICAÇÃO DE DOCUMENTOS DIGITAIS POR SISTEMAS CRIPTOGRÁFICOS DE CHAVE PÚBLICA

RESUMO

Autenticar um documento é permitir que sua origem e integridade sejam verificados de maneira segura. Quando um documento está escrito em papel, formas tradicionais de autenticação incluem assinaturas, marcas e produtos especiais.

No universo digital, o problema de autenticação se torna difícil pela facilidade de copiar e alterar os dados sem que se possa detectar a falsificação no futuro. Para resolver esse problema e permitir a autenticação de documentos e mensagens digitais foram desenvolvidos métodos baseados em criptografia. Utilizando esses métodos, podem ser estabelecidos contratos e mensagens podem ser trocadas de forma segura, sem que seja possível falsificar uma mensagem ou, conseqüentemente, negar sua autoria.

Neste trabalho apresentamos uma introdução sobre os métodos de autenticação baseados em sistema criptográficos de chave pública. Utilizando assinaturas digitais, esses métodos apresentam características vantajosas sobre métodos baseados sobre sistemas criptográficos tradicionais, i.e., de chave secreta. Dois métodos de assinatura são discutidos, o RSA e o DSA, como exemplo das principais técnicas de assinatura sendo utilizadas atualmente.

Sumário

1. INTRODUÇÃO

O uso generalizado de computadores e suas redes para guarda e transmissão de informações exige cada vez mais que a informação possa ser verificada quanto a sua integridade e origem, i.e., quanto a sua autenticidade. Documentos escritos sobre uma folha de papel podem ser garantidos autênticos por meio de marcas especiais pertencentes ao documento, como uma assinatura ou o uso de técnicas especiais de impressão, porém, um documento transmitido ou arquivado como uma seqüência de dígitos binários está sujeito a várias modificações que, a priori, não podem ser detectadas.

A soluções tradicionais para garantir segurança e autenticidade de mensagens sujeitas a ataques sempre envolveram técnicas de escrita secreta. Usualmente destinadas a manter o conteúdo de uma mensagem desconhecido de um inimigo, estas técnicas também asseguravam a origem e a integridade da mensagem.

Entre as várias técnicas de escrita secreta conhecidas, a mais adequada a aplicação em computadores são as cifras. Uma cifra implica em um método, ou sistema, de escrita secreta que é ilimitado no seu uso e pelo qual deve ser possível transformar qualquer mensagem, sem consideração sobre linguagem e tamanho, para uma forma não compreensível, chamada criptograma. O processo de transformação da mensagem em claro em criptograma se chama cifragem, enquanto o método de transformar o criptograma na mensagem em claro original se chama decifragem. Tanto para cifragem quanto para decifragem é necessário um segundo parâmetro: a chave. À ciência que estuda as formas de ocultar um texto chama-se criptografia, a ciência que estuda como quebrar um sistema criptográfico chamamos criptoanálise, a reunião das duas ciências forma o campo chamado criptologia.

1.1. Sistemas Criptográficos Simétricos

Tendo em vista a característica das cifras de garantir, enquanto escondiam o conteúdo da mensagem, a origem e a integridade da mesma, ficou claro que seriam uma forma de autenticação de documentos. Alguns métodos de autenticação que não serão tratados nesse artigo utilizam métodos tradicionais de cifragem, os sistemas criptográficos simétricos, onde a chave de cifragem é a mesma da chave de decifragem, como o sistema Kerberos, do MIT [36].

Em um método de autenticação utilizando um sistema criptográfico simétrico, o princípio de autenticação é descrito por: "Se uma entidade pode cifrar corretamente um mensagem utilizando uma chave que o verificador acreditar ser conhecida apenas pela entidade com a identidade reclamada, este ato constitui prova suficiente de identidade." [36]

Porém, alguns problemas se colocaram no uso de cifras para autenticar documentos. Entre eles, a necessidades de manter o conteúdo do documento em aberto e a necessidade de permitir que qualquer um, a qualquer momento, certifique a autenticidade do documento ou do remetente. Sistemas criptográficos simétricos apresentam desvantagens na identificação do emissor da mensagem, pois necessitam que exista uma chave para cada par de usuários ou uma terceira entidade que conheça todas as chaves.

1.2. Sistemas Criptográficos Assimétricos

A solução para garantir a autenticidade de mensagens e a identidade do autor apareceu com o advento de sistemas criptográficos de chave pública, ou sistemas criptográficos assimétricos [7], que possuem chaves distintas de cifragem e decifragem, que permitem que um documento seja cifrado apenas por uma pessoa, mas decifrado por várias, ou vice-versa. O originador do documento usa uma chave privada para realizar a cifragem, e publica em um diretório uma chave pública correspondente a sua chave privada. Assim, todos os interessados podem utilizar sua chave pública para determinar a origem do documento. Sistemas de chave pública devem ter sua segurança dependente apenas da segurança da chave privada. Assim, quanto mais resistente for a chave privada a tentativas de ataque, i.e., tentativas de falsificação, mais certeza pode se ter da autenticidade do documento.

Apesar da necessidade de se manter o documento em aberto ser resolvida pelo uso da chave pública, um método mais interessante é a utilização de um resumo da mensagem. Um resumo de mensagem é o resultado da aplicação de uma função unidirecional sobre a mensagem original, gerando uma segunda mensagem, de tamanho muito menor que a original. Esses tipos de funções garantem que a partir da mensagem pode se encontrar facilmente seu resumo, porém para encontrar a mensagem a partir do resumo é necessária uma busca exaustiva. Aplicando a cifragem de chave pública sobre o resumo da mensagem obtém-se uma assinatura digital.

Devido as características especiais do sistema de chave pública e dos resumos de mensagens, uma assinatura digital garante a origem e integridade da mensagem e não pode ser negada.

O princípio básico de sistemas de autenticação baseados em métodos de chave pública é: “Se uma entidade pode assinar corretamente um mensagem utilizando a chave privada da identidade reclamada, então esse ato constitui prova suficiente de autenticação [36].”

1.3. Usos Específicos

Vários são os usos específicos da autenticação de mensagens digitais, entre eles podemos citar: autenticação para sistemas distribuídos [36], assinatura de contratos, transferência de fundos e assinatura cega [4].

1.3.1. Autenticação em sistemas distribuídos

Woo e Lam [36] descrevem motivação e sistemas para autenticação em sistemas computacionais distribuídos. Um sistema distribuído é submetido a várias ameaças, entre elas o comprometimento das comunicações. Três são as ameaças as comunicações em um ambiente distribuído: escuta clandestina, modificação arbitrária do de mensagens e replay de mensagens antigas (uma união das anteriores).

Para evitar esses três ataques, também três formas de autenticação são necessárias: autenticação de integridade da mensagem, autenticação da origem da mensagem e autenticação da identidade de um entidade, i.e., verificação que uma entidade possui a identidade por ela reclamada. O segundo caso pode ser visto como um subproblema do terceiro.

1.3.2. Assinaturas Cegas

Uma assinatura cega [4] é uma extensão de uma assinatura digital que inclui privacidade. Em um sistema de assinatura digital, é possível criar um relatório sobre as atividades das entidades que realizam a assinatura de uma seqüência de documentos. Em uma assinatura cega,

é impossível recuperar os passos do emissor das mensagens. O artigo de David Chaum [4] explica várias aplicações de assinaturas cegas e chama a atenção do uso de técnicas de autenticação em processadores de capacidade computacional limitada, que estão sendo instalados em cartões inteligentes (smart cards).

1.3.3. Serviços de estampagem de tempo

Um serviço de estampagem de tempo fornece certificados associando um documento com uma data de uma forma criptograficamente segura. Esse certificado pode ser utilizado mais tarde para provar que determinado documento existia no tempo de sua certificação. Uma das formas sendo utilizadas atualmente para provar a existência de um documento em uma certa data é publicar a assinatura digital desse documento em um jornal.

1.4. Apresentação

No capítulo 2 será feita uma pequena introdução a criptologia. No capítulo 3 serão apresentados o que são assinaturas digitais e os métodos que a utilizam. A seguir, serão apresentados alguns algoritmos para os métodos de cifragem e resumo, nos capítulos 4 e 5. A conclusão apresenta um resumo do trabalho apresentado e uma proposta de continuação do trabalho.

2.1. Sistemas Criptográficos

Dois são os objetivos básicos da criptografia: garantir o segredo e a autenticidade de uma mensagem. Um Sistema Criptográfico [6] possui cinco componentes:

1. um espaço de mensagens em claro, p;
2. um espaço de mensagens cifradas, f;
3. um espaço de chaves, n;
4. uma família de transformações de cifragem, Ek: p D f, onde k U n e
5. uma família de transformações de decifragem, Dk: f D p, onde k U n.

Para uma chave k, D_k é a inversa de E_k, logo, para qualquer mensagem dada M, M U p , $D_k(E_k(M)) = M$.

2.1.1. Sistemas criptográficos assimétricos

Em um sistema criptográfico assimétrico, ou de duas chaves, as chaves de cifragem e decifragem são diferentes, de forma que seja computacionalmente impossível determinar uma a partir do conhecimento da outra. A autenticação é obtida mantendo E_k secreta. Este conceito foi introduzido em [7] e foi chamado de cifragem por chave pública.

Nos sistemas de autenticação por chave pública, cada usuário A tem uma transformação privada de cifragem E_A e uma transformação pública de decifragem D_k. A transformação privada é descrita por um chave privada (ou secreta) e a transformação pública por uma chave pública, derivada a partir da chave provada por uma transformação unidirecional.

Para compreender melhor a notação apresentada em função dos sistemas de chave pública, basta considerar, como Bellare e Micali em [3] que o espaço de chaves n é formado de pares ordenados (P_k,S_k), onde P_k é a chave pública e S_k é a chave secreta. Outra opção foi usada por Simmons em [34], que define um sistema criptográfico com dois espaços de chaves, n e n', um para a chave de cifragem, k, outro para a chave de decifragem, k'. Nesse caso, se k = k', o sistema é simétrico, se k ≠ k', assimétrico.

2.2. Autenticidade

Um sistema criptográfico para garantir autenticidade exige que um criptoanalista não seja capaz de trocar um texto cifrado C por um texto cifrado C' sem detecção. Formalmente, ele deve cumprir dois requisitos:

1. Deve ser computacionalmente impossível para um criptoanalista sistematicamente determinar a transformação de cifragem E_k dado C, mesmo que o texto em claro M seja conhecido.
2. Deve ser computacionalmente impossível para um criptoanalista sistematicamente encontrar uma mensagem cifrada C' tal que $D_k(C')$ é um texto em claro válido no conjunto p.

O primeiro requisito assegura que um criptoanalista não pode sistematicamente determinar a transformação de cifragem, logo, ele é incapaz de cifrar um texto em claro M' e substituir o texto cifrado falso $C'=E_k(M')$ por C. O segundo requisito assegura que o criptoanalista

não pode encontrar um texto cifrado C' que pode ser decifrado para um texto em claro significativo sem a transformação de decifragem.

A autenticação (ao contrário do segredo) requer apenas que a transformação de cifragem E_k seja mantida secreta, permitindo que D_k seja revelada, caso não facilite o descobrimento de E_k.

2.3. Criptoanálise

A própria necessidade de utilizar sistemas criptográficos já indica que existe uma ou mais entidades interessadas em interferir em um canal de comunicação com a finalidade de alterar dados em proveito próprio. Estas entidades, normalmente chamadas de atacantes, utilizam técnicas de criptoanálise.

A segurança de um sistema criptográfico é medida pelo tipo de ataque que ele pode sofrer sem que seja quebrado, i.e., sem que a segurança seja comprometida. Considera-se que a forma mais segura de sistema criptográfico deve resistir a ataques mesmo quando todos seus algoritmos são publicados e os usuários do sistema colaboram com o criptoanalista, sendo mantida em segredo apenas a chave.

Existem dois tipos básicos de ataques: ataques pela chave pública e ataques por mensagens. No ataque pela chave, o inimigo conhece apenas a chave pública do usuário. Nos ataques por mensagens, vários esquemas podem ser montados, dependendo da forma que o atacante tem para utilizar as mensagens do usuário.

Dos vários tipos de ataques por mensagens possíveis, o mais geral é o método do ataque adaptativo escolhido. Nesse ataque, o criptoanalista utiliza o "assinador" A para obter assinaturas de mensagens da sua escolha, sendo permitido que ele escolha as mensagens não apenas em função da chave pública de A, mas também em função das respostas fornecidas por A. No método do texto em claro conhecido são dados ao criptoanalista pares texto em claro e criptograma com a mesma chave. No método do texto em claro escolhido o criptoanalista pode escolher quais textos em claro devem ser cifrados com uma determinada chave (mas não pode escolhe-los em função da resposta).

Os procedimentos de criptoanálise mais ingênuos são os de força bruta, a busca exaustiva ou a busca em tabela. No processo de busca exaustiva o atacante utiliza o método do texto em claro conhecido para experimentar todas as chaves possíveis. A complexidade de tempo deste método é O(n), onde n é o tamanho da chave, a complexidade de espaço é O(1). No processo da busca em tabela utiliza-se o método do texto em claro escolhido. Para um texto em claro escolhido são pré-computados todos os criptogramas relativos a todas as chaves, sendo indexados pelo criptograma. Dado um criptograma, a descoberta da chave acontece em um tempo proporcional a O(1), mas o espaço ocupado é proporcional a O(n).

Todo sistema de chave criptográfico de chave pública pode ser quebrado não deterministicamente em tempo polinomial e, desde que, minimamente, não deve ser possível que sejam quebrados deterministicamente em tempo polinomial, uma prova que um determinado sistema é seguro equivale a provar que $P \neq NP$ [10]. É importante deixar clara a dificuldade de se criar um sistema criptográfico comprovadamente seguro, já que o problema equivalente é um dos mais importantes problemas em aberto da matemática.

3. ASSINATURAS DIGITAIS

“Assinaturas digitais funcionam para documentos digitais assim como assinaturas comuns funcionam para documentos impressos: a assinatura é uma mensagem não falsificável garantindo que uma certa pessoa ou entidade escreveu ou está de acordo com o documento no qual a assinatura está colocada.” [9]

“Um algoritmo de assinatura é um algoritmo que transforma uma mensagem de qualquer comprimento e uma chave privada em uma assinatura, de tal modo que seja computacionalmente não realizável encontrar duas mensagens com a mesma assinatura, encontrar uma mensagem com uma assinatura pré-determinada ou encontrar a assinatura para uma mensagem sem utilizar a chave privada.”[14]

Como tal, a assinatura digital permite que qualquer pessoa que leia uma mensagem se certifique que ela realmente foi assinada pelo originador da assinatura (garantia de origem) e de que a mensagem não foi modificada (garantia de integridade). Além disso, assinaturas digitais não podem ser repudiadas, isto é, o originador da assinatura não pode, após assina-la, deixar de cumprir as obrigações determinadas pela assinatura utilizando a afirmação que a assinatura teria sido falsificada.

Uma assinatura digital funciona da seguinte forma: Alice[1], para assinar uma mensagem, realiza uma algoritmo envolvendo simultaneamente sua chave secreta e a mensagem propriamente dita, enviando o resultado, i.e., a assinatura (Fig. 1), junto com a mensagem. Bob, para verificar a assinatura, realiza um segundo algoritmo envolvendo a mensagem recebida, a assinatura e a chave pública de Alice. Se o resultado do algoritmo aplicado por Bob possui alguma propriedade pré-definida (Fig. 2), então a mensagem é considerada genuína, caso contrário, é considerada alterada e descartada.

Logo, um sistema de autenticação por assinatura digital pode ser definido informalmente por duas funções, uma definindo um método de assinatura de tal modo que a falsificação seja impossível, e outra, definindo um método de certificação.

Atualmente existem vários sistemas de assinaturas digital, porém apenas um tem larga aceitação, baseado no sistema de chave pública conhecido como RSA. Recentemente uma agência do governo americano (NIST), auxiliada pela NSA, publicou o Digital Signature Schema (DSS), um novo padrão de assinatura digital que vem sendo alvo de muitas críticas, particularmente pelos possuidores da patente do RSA.

---

[1] Em criptografia, Alice, Bob e Charlie são os nomes padrões utilizados para exemplificar os usuários e suspostos atacantes para um método ou sistema.
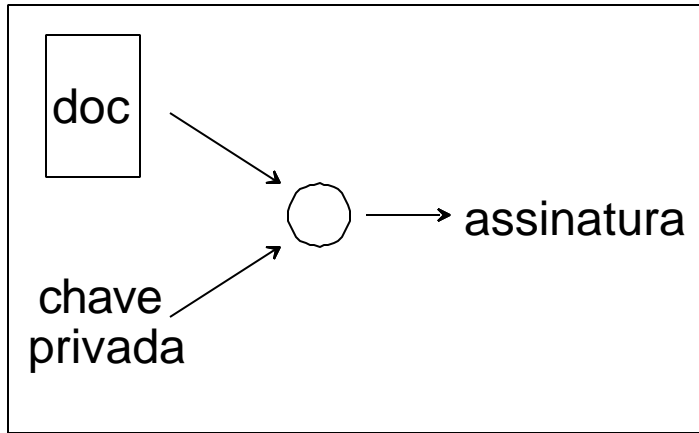
Fig. 1: A assinatura é gerada por meio de uma função que tem como parâmetros a chave privada e o documento.
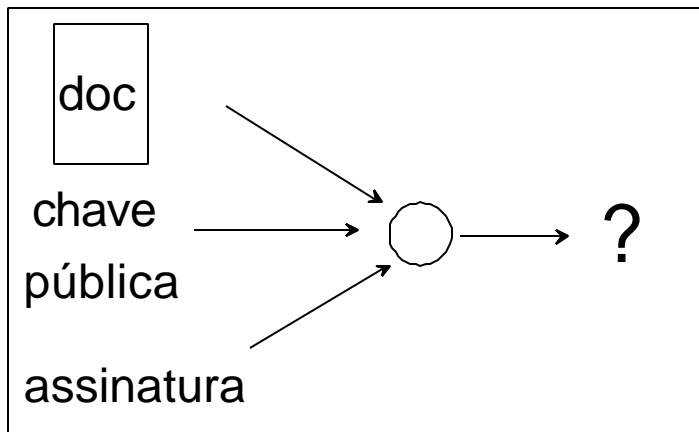
Fig 2: A verificação tem como parâmetros a chave pública, o documento e assinatura.

3.1. Sistemas de chave pública

O conceito de sistema criptográfico de chave pública foi apresentado pela primeira vez por Diffie e Hellman em 1976 [7], porém eles não conseguiram apresentar um método implementando a idéia.

Em seu artigo, Diffie e Hellman propuseram que sistemas de chave pública fossem criados a partir de funções com segredo. Uma função com segredo é uma extensão do conceito de funções unidirecionais. Uma função é dita unidirecional se é computacionalmente fácil de realizar, mas computacionalmente difícil de inverter. Uma função com segredo é uma função com um

parâmetro associado com o qual é fácil realizar a inversão, mas que, sem esse parâmetro, aparenta ser unidirecional.

Nesse trabalho apresentaremos dos algoritmos de assinatura baseados na idéia de utilizar funções para as quais a manutenção de um certo valor em segredo causa uma dificuldade computacional na computação da função inversa. Ambos são construídos sobre problemas matemáticos computacionalmente difíceis. A maioria dos métodos em uso utilizam um desses dois problemas ou uma modificação que os torna mais difíceis. Inicialmente também se utilizava o problema da mochila, mas as principais aplicações deste foram demonstradas inseguras.

3.1.1. O problema da fatorização

Sistemas de chave pública baseados no problema da fatorização utilizam a característica que multiplicar dois números primos é uma tarefa fácil, enquanto fatorar um número grande em dois números primos é uma tarefa computacionalmente muito mais difícil.

Um exemplo de sistema baseado na fatorização em número primos é o RSA, um sistema de chave pública para cifragem e autenticação inventado em 1977 por Ron Rivest, Adi Shamir e Leonard Adleman [24]. O RSA é patenteado pela RSA Data Security, Inc.

É importante notar que, quanto mais avançadas forem as técnicas de fatorização, é razoável admitir que não haverá modificação na utilidade destes algoritmos, pois os primos são gerados utilizando as mesmas técnicas. Logo, quanto mais fácil for fatorar um número de um tamanho específico, mais fácil será gerar primos de tamanhos maiores.

Considerando que a busca de fatores primos é um dos problemas mais antigos da matemática, sua utilização para um problema prático é considerada de grande elegância.

3.1.2. O problema do logaritmo discreto

Na sua forma mais simples, o sistema do logaritmo discreto tenta descobrir o expoente x na fórmula $y = g^x \bmod p$. A segurança dos sistemas depende da dificuldade de determinar o valor de x.

O Algoritmos de ElGamal e do DSA são baseados no problema do logaritmo discreto. O primeiro foi publicado por ElGamal em [8], enquanto o segundo foi publicado pelo NIST em [20]. Modificações desse problema utilizam a exponencial discreta, que parece ser mais segura que o problema original.

3.2. Alguns sistemas de autenticação usando RSA

Sendo o RSA o algoritmo mais utilizado atualmente para autenticação, é importante apresentar vários sistemas que o utilizam.

3.2.1. Internet Privacy Enhanced Mail (PEM)

O Internet PEM é um padrão projetado, proposto mas ainda não adotado oficialmente pelo Internet Activities Board para permitir a troca de mensagens eletrônicas (mail) seguras pela Internet. PEM inclui padrões para cifragem, autenticação e gerência de chaves, sendo um padrão inclusivo, do qual fazem parte sistemas simétricos, assimétricos e suporte para várias ferramentas criptográficas, sendo que a função de hash (resumo), cifragem e assinatura são especificadas no cabeçalho da mensagem.

O padrão PEM cita algoritmos considerados seguros, permitindo a posterior inclusão de novos métodos. Atualmente ele cita o RSA e o DES [17].

3.2.2. Assinatura digital no Public-Key Cryptografic System (PKCS)

O PKCS é uma coleção de padrões [14] proposta pela RSA Data Security, Inc., utilizando sistemas assimétricos para realizar as seguintes tarefas:

- assinatura digital,
- envelopamento digital, onde uma mensagem é "selada" de tal forma que só pode ser lida por um destinatário especificado,
- certificação digital, onde uma autoridade de certificação assina uma mensagem especial contendo o nome de algum usuário e sua chave pública, de tal forma que qualquer um possa ter confiança na chave pública desse usuário e
- concordância de chave, onde duas entidades, sem arranjo prévio, trocam mensagem de tal forma a concordar com uma chave privada conhecida apenas por elas, que pode ser usada no futuro como chave para um sistema simétrico.

Tendo sido realizado pela empresa possuidora da patente do RSA, não é de se espantar que o sistema de chave pública descrito nesses padrões seja o próprio RSA.

Os objetivos globais do PKCS são manter compatibilidade com o Internet PEM, estender o Internet PEM para lidar com qualquer tipo de dados e tratar um número maior de atividades e servir como proposta para ser parte dos padrões OSI.

Tanto o PKCS quanto o PEM utilizam os métodos de resumo conhecidos como MD2 [15], MD4 [22] e MD5 [23].

3.2.3. SPX

O SPX é um sistema de autenticação para sistemas distribuídos que faz parte do Digital Distributed System Security Architecture [4]. Tem a mesma função do Kerberos, porém utiliza software baseado no RSA.

3.3. O Digital Signature Standard (DSS)

Em agosto de 1991 o National Institute of Standards and Technology (NIST), guiado pela National Security Agency, lançou a proposta de um padrão de assinatura digital, o Digital Signature Algorithm (DSA), baseado nos algoritmos de Schnorr [33] e ElGamal [8].

Este padrão foi fortemente atacado e considerado por muitos "fraco", "irracional" e "potencialmente perigoso". Como o padrão "de facto" da indústria de computação é o RSA, fatores econômicos multiplicaram as críticas aos fatores técnicos. Depois das críticas, ao menos uma modificação simples, porém de grande importância foi realizada no proposta.

É importante notar que o DSA só pode ser utilizado para autenticação, e não para cifragem ou troca de chaves.

Como alternativa ao DSA, a Moëbius Encription Technology propôs o Moëbius Digital Signature Scheme[2] .

O DSS deve utilizar o método de resumo conhecido como SHS, proposto quase que simultaneamente pelo NIST.

4. Algoritmos para o Sistema de Chave Pública

Nesse capítulo são descritos os algoritmos de chave pública mais discutidos atualmente: o RSA e o DSA.

4.1. RSA

O RSA se inicia com dois grandes números primos, p e q. Dados esses números, encontra-se seu produto, n=pq, denominado módulo. Um terceiro número e, menor que n e relativamente primo com (p-1)(q-1) é escolhido e seu inverso mod (p-1)(q-1), d, calculado. Isto significa que ed 4 1 mod (p-1)(q-1) ; e e d são chamados expoentes público e privado, respectivamente. A chave pública é o par (n,e), enquanto a chave privada é d. Os fatores p e q devem ser mantidos secretos ou destruídos.

Se Alice deseja enviar um documento assinado m para Bob, ela cria a assinatura s fazendo a operação $s = m^d$ mod n, onde d e n pertencem as chaves de Alice. Alice envia então s e m para Bob. Para verificar a assinatura, Bob realiza a operação s^e mod n. Se o resultado for m, a mensagem é autêntica. Isso pode ser verificado pelas equações abaixo:

1. $s = m^d$ mod n
2. $m = s^e$ mod n
3. de = 1 mod ((p-1)(q-1))

4. $m = m^{de}$ mod n — de 1 e 2 ([2])
5. Como n=pq, p e q primos, @(n) = (p-1)(q-1) ([3])
6. de = 1 mod @(n) — de 3 e 5

7. $m = m^1$ mod n — de 4 e 6 ([4])

4.1.1. Ataques ao RSA

O ataque mais grave ao RSA é descobrir a chave privada correspondente a uma chave pública. O caminho óbvio é tentar fatorar o número n e descobrir os fatores p e q. Se o RSA for equivalente ao problema da fatoração, esse é o único ataque possível, porém isso ainda não foi provado e é razoável admitir que pode existir outro ataque ao RSA que não tente uma fatoração direta de n.

---

[2] $s - m^d = an, m - s^e = bn, s = m^d + an, m - (m^d+an)^e = bn, m - m^{de} - n (...) = bn, m - m^{de} = cn$

[3] Onde @(*n*) é a função de Euler para n e representa a quantidade de números menores e primos em comum com *n*.

[4] Pelo teorema de Euler, se mdc(a,n)=1, então $a^{@(n)} = 1$ *mod n*. Isso permite provar que se *a = b mod* @(*n*), então $p^a = c$ *mod n* L $p^b = c$ *mod n*. Basicamente, $a = k@(n) + b$, $p^a = c$ *mod n*, $p^{k@(n)+b} = c$ *mod n*, $p^b = c$ *mod n*. Para maiores explicações, ver [1,5,12].

Uma das características do RSA é que ele um sistema multiplicativo. Uma função é dita multiplicativa quando dado f(x) e f(y), calcular f(xy) é fácil. Isso significa que o RSA é suscetível a ataques por texto em claro escolhido.

4.1.2. Escolha dos valores

Até pouco tempo se considerava que os primos p e q deviam ser primos fortes. Primos fortes são números primos com características especiais que os tornam mais difíceis de serem encontrados por certas técnicas de fatoração. Porém recente descobertas em técnicas de fatoração que apresentam performance semelhante com números primos normais e números primos fortes acabaram com essa necessidade.

O tamanho do módulo n determina a dificuldade de fatorização e descoberta de p e q. Rivest [25] calcula que uma chave de 512 bits pode ser fatorado atualmente com um custo de US$8.2 milhões. Ele sugere que pessoas com dados muito importantes usem chaves de 700 a 800 bits, enquanto uma autoridade de certificação pode usar uma chave de 1000 bits ou mais. Os números p e q devem ter tamanho aproximadamente iguais, logo se n tiver 512 bits, p e q devem ter 255 e 257 bits.

Normalmente, e é escolhido como 3 ou 2^{16} - 1, que é o quinto primo de Fermat[5]. A escolha de e parece não afetar a segurança do método, e o uso do número 3 permite implementações velozes..

4.1.3. RSA na prática

Na prática, ao invés de assinar a mensagem completa, o primeiro passo de quem quer utilizar o RSA para assinaturas digitais é aplicar sobre a mensagem um algoritmo de hashing. Esse algoritmo cria uma "impressão digital" da mensagem, i.e., um número, que então é cifrado com a chave privada.

Verifica-se então que o algoritmo de hashing passa a ser item importante na segurança da assinatura, pois deve ser impossível realizar um ataque com o objetivo de encontrar outra mensagem com o mesmo hashing (resumo). Vários algoritmos foram propostos, sendo que alguns deles são descritos no próximo capítulo.

4.1.4. Certificados

Um certificado é uma prova de identidade fornecida por uma entidade conhecida e confiável, que garante que uma assinatura pertence a entidade que a reclama. Certificados são fornecidos por uma Autoridade Certificadora (AC). As chaves privadas das ACs são de extrema importância e devem ser mantidas em uma Unidade de Assinatura de Certificados, uma caixa de alta segurança que destrói seu conteúdo se aberta.

Como uma AC pode ser alvo de um ataque intensivo, sua chave deve ser muito longa, provavelmente 1000 bits ou mais.

---

[5]Números de Fermat são números na forma $2^{2^n} + 1$, os quais Fermat conjecturou que seriam todos primos em 1640, devido a sequência: F_0=3, F_1=5, F_2=17, F_3=257, F_4=65537. Porém ,em 1732, Euler provou que F_5 não é primo, pois F_5=4294967297 = (641)*(6700417), derrubando a conjectura.

4.2. DSA

O Digital Signature Algorithm (DSA) é um algoritmo de assinatura digital (e não de cifragem) utilizado em conjunto com uma função de hash H, não especificada no padrão. A base de partida do algoritmo é um conjunto de parâmetros:

p = um módulo primo, $2^{511} < p < 2^{512}$,
q = um primo divisor de p-1, onde $2^{159} < q < 2^{160}$,
g = $h^{(p-1)/q}$ mod p, onde h é qualquer inteiro positivo menor que p tal que g > 1,
x = um inteiro positivo menor que q,
y = g^x mod p,
m = a mensagem a ser assinada,
k = um inteiro aleatório menor que q,
H = uma função de hashing unidirecional.

Os inteiros p,q e g podem ser públicos e comuns a um grupo de usuários. A chave privada é x e a chave pública y. x e k precisam ser mantidos secretos, mas k pode ser trocado a cada assinatura.

4.2.1. Assinatura e Verificação

Para assinar uma mensagem k, o usuário escolhe um k aleatório e calcula:

$r = (g^k \bmod p) \bmod q$
$s = (k^{-1}(H(m) + xr)) \bmod q$

onde k^{-1} é o inverso multiplicativo de k, mod q; isto é:

$k^{-1}k \bmod q = 1$
e $0 < k^{-1} < q$

Os valores r e s são a assinatura da mensagem, sendo transmitidos junto com m.

Antes de verificar a mensagem assinada, o recipiente da mensagem recebe de maneira autenticada os valores p, q, g e a chave pública do emissor.

Sejam m', r', e s', as versões recebidas de m,r e s, respectivamente, e seja y a chave pública do emissor. Para verificar a assinatura o recipiente primeiro verifica se:

$0 < r' < q$, e
$0 < s' < q$

Se alguma dessas condições foi violada, assinatura é rejeitada. Caso positivo, o recipiente calcula:

$w = (s')^{-1} \bmod q$
$u_1 = ((H(m'))w) \bmod q$
$u_2 = ((r')w) \bmod q$
$v = (((g)^{u_1} (y)^{u_2}) \bmod p) \bmod q$

Se v = r'', a assinatura foi verificada e o recipiente pode ter uma confiança alta de que a mensagem recebida foi enviada pela entidade que possui a chave privada x correspondente a y. Caso contrário, a mensagem foi comprometida e deve ser considerada inválida.

A prova que v=r' pode ser encontrada em [20] e, devido a complexidade, foge do escopo desse trabalho.

4.2.2. Outras características do processo

Em [20] outras características do processo são apresentadas. Essas características não fazem parte do padrão, mas servem de suporte ao mesmo. Entre elas estão: a prova que v = r' , sugestões de métodos para geração dos parâmetros e duas pequenas discussões sobre a geração de números aleatórios e implementações de matemática modular em software e hardware.

4.2.3. Críticas ao DSA

As críticas mais importantes ao DSA partiram de Ronald L. Rivest [25] (O R do RSA) e Martin Hellman [11]. Eles apontam, entre outros, os seguintes fatos:

- a chave é muito curta (512 bits),
- k pode ser fácil de descobrir,
- o processo de seleção foi, no mínimo, secreto ou falho,
- existe um problema de patentes, principalmente com o método de Schnorr e com a Public Key Partner's, possuidora da patente do modelo de criptografia por chave pública,
- houve uma inversão das expectativas do tempo para assinar e verificar na proposta,
- o RSA já é um padrão para 2/3 das companhias americanas, incluindo IBM, Apple, Digital e outras, e um padrão de entidades mundiais, incluindo ISO, CCITT e Internet,
- o DSS não inclui um mecanismo de troca de chaves para o estabelecimento de comunicação por um sistema criptográfico simétrico, ou ainda um mecanismo de envelope e
- o sistema não foi estudado por tempo suficiente para garantir sua segurança (enquanto o RSA já sofreu 15 anos de ataques).

4.2.4. Respostas as Críticas

A mais importante resposta as críticas ao DSA foi a modificação do tamanho do módulo p, que passou a poder variar entre 512 e 1024 bits.

As outras respostas envolvem basicamente argumentos políticos e podem ser encontradas em [19].

4.3. Outros Algoritmos

Vários outros algoritmos foram propostos para a utilização em sistemas de chave pública. Merkle e Hellman [12] apresentaram um algoritmo baseado no problema da "mochila ", um problema NP-completo, que demonstrou ser inseguro. ElGamal[8] propôs um algoritmo baseado no problema do logaritmo discreto. Schnorr [33] propôs uma modificação ao algoritmo de ElGamal que serviu de base para o desenvolvimento para o DSA. Em 1980, Williams [35] apresentou uma modificação ao RSA que é vulnerável a um ataque por cifras selecionadas. Ainda, como crítica ao DSA, foi proposto um outro sistema chamado Moëbius[2], cuja segurança é baseada na solução da equação $x^{f(x)}$ = -.

5. ALGORITMOS PARA OS RESUMOS DE MENSAGENS

Nesse capítulo discutiremos alguns algoritmos para resumo de mensagem. O mais simples deles, MD2, será completamente descrito, de forma a criar a idéia de como funciona um algoritmo de hashing para resumo de mensagens.

5.1. Funções de Hash e Resumos de Mensagens

Uma função de hash é uma computação que a partir de uma entrada de tamanho variável produz uma string de tamanho fixo, que é chamado valor de hash. Se uma função de hash é também uma função unidirecional, o resultado é chamado resumo de mensagem. A idéia é que um resumo de mensagem representa concisamente um mensagem mais longa ou documento que serviu de entrada para o seu cálculo[9]. Isso faz com que a computação da assinatura se torne muito mais eficiente e permite que uma assinatura seja revelada sem que o documento o seja (permitindo assim a utilização de sistemas de estampagem de tempo).

Quando utilizada para autenticação de mensagens, uma função de hash deve ter algumas propriedades específicas: deve ser impossível achar uma mensagem que possua um resumo determinado ou encontrar duas mensagens com o mesmo resumo. A habilidade de realizar um desses atos permitiria a um atacante substituir uma mensagem por outra ou a uma entidade negar a assinatura de uma mensagem, dizendo que ela foi trocada.

Para se prevenir de ataques baseados em busca exaustiva,. uma função de resumo de mensagem tem que produzir valores de hash de um tamanho mínimo. Por exemplo, um resumo com 100 bits levaria 2^{100} tentativas, em média, para alcançar um valor específico, ou 2^{50} tentativas para encontrar duas mensagens com o mesmo resumo.

MD2[15], MD4 [22] e MD5[23] são algoritmos propostos por Rivest para produzir resumos de mensagens. MD2 é o mais lento, MD4 o mais rápido e o utilizado pelo SNMP (Secure Network Management Protocol], MD5 é uma forma mais segura do MD4, porém mais lenta.

As funções MD podem ser atacadas em 2^{64} operações, o que é equivalente a quebrar um RSA com chave de 512 bits. Entidades que necessitam de segurança maior devem utilizar outro algoritmo, como o SHS, com um resumo de 160 bits, ou uma modificação do MD4 para gerar resumos de 256 bits.

5.1.1. Outros Métodos

Outros métodos foram propostos como função de hash, entre eles o N-hash e o Snefru, que demonstraram fraqueza com relação a um ataque por criptoanálise diferencial.

5.2. MD2

O algoritmo de resumo de mensagem MD2 recebe uma mensagem de tamanho arbitrário e produz um resumo de 128 bits. Considera-se ser computacionalmente impossível produzir duas mensagens com o mesmo resumo. O algoritmo é destinado para sistemas de assinatura digital, onde uma mensagem deve ser comprimida de uma forma segura antes de ser assinada com uma chave privada segundo um sistema criptográfico assimétrico como o RSA.

O algoritmo inicia com uma mensagem de b, b um inteiro não negativo, bytes como entrada, para qual deve ser encontrado o resumo. Imaginamos que a mensagem seja escrita da seguinte forma:

$m_0 , m_1 , \dots , m_{(b-1)}$

os cinco passos seguintes são executados para computar o resumo da mensagem.

5.2.1. passo 1: adição de bytes extras

A mensagem é completada (estendida) de forma que seu tamanho, em bytes, seja um múltiplo de 16. A extensão é sempre realizada, mesmo que o tamanho da mensagem já seja um múltiplo de 16. Ao menos 1 byte e no máximo 16 bytes são utilizados para completar a mensagem.

A extensão é realizada colocando-se i bytes de valor i no fim da mensagem.

Nesse ponto, a mensagem resultante pode ser denotada como M[0 ... N-1], onde N é um múltiplo de 16.

5.2.2. passo 2: concatenação do checksum

Ao resultado do passo anterior, concatena-se um checksum de 16 bytes. Esse passo usa uma permutação pseudo-aleatórioa de 256 bytes baseada nos dígitos de ;. Seja S[i] o i-ésimo elemento da tabela com essa permutação (que pode ser encontrada em [15]).

Para isso, deve ser executado o seguinte algoritmo:

```
1.  /* inicialização */
2.  Para i=0 até 15
3.      C[i]=0
4.  fim

5.  L=0

6.  Para i=0 até N/16-1
7.      Para j=0 até 15
8.          c = M[i*16+j]
9.          C[j]= S[c xor L]
10.         L = C[j]
11.     fim
12.fim
```

5.2.3. passo 3: inicialização do buffer MD

Um buffer X, de 48 bytes, é inicializado em zero,

5.2.4. passo 4: processamento em blocos de 16 bytes

Executar o seguinte algoritmo:

```
1.  /* processar em blocos de 16 bytes */
2.  Para i=0 até N'/16-1
3.      Para j=0 até 15
4.          X[j+16]= M[i*16+j]
```

```
5.        X[j+32]= X[16+j] xor X[j]
6.     fim

7.     Para j=0 até 17
8.        Para k=0 até 47
9.           t = X[k] xor S[t]
10.          X[k]= X[k] xor S[t]
11.       fim
12.       t=(t+j) mod 256
13.    fim
14.fim
```

5.2.5. passo 5: saída

O resumo de mensagem produzido é X[0],...,X[15]. (16*8 bits = 128 bits).

5.3. Outros métodos

5.3.1. MD4 e MD5

Tanto o MD4 quanto o MD5 seguem os mesmos passos de MD2, porém com algoritmos diferentes. Ao invés de utilizarem o checksum, eles utilizam o tamanho da mensagem no passo 2. Os passos 4 de MD4 e MD5 produzem funções mais seguras. A descrição completa dos algoritmos e implementações de referência podem ser encontradas em [22] e [23].

5.3.2. SHS

O Secure Hash Standard [SHS] proposta pelo NIST para uso com o DSA. É uma função de resumo de mensagem que fornece um resumo de 160 bits, semelhante estruturalmente a MD4 e MD5. Sua implementação é 25% mas lenta que MD5, porém pode ser mais segura, pois seus resumos são 25% maiores que os de MD4 e MD5. Essa função ainda não foi adotada formalmente como padrão.

6. CONCLUSÃO

Foram apresentados os métodos mais importantes de assinaturas digitais e resumo de mensagens, com uma pequena abordagem do problema atual causado pela proposta da NIST.

Com esse trabalho foi desenvolvida um compreensão geral do mecanismo de assinatura digital que permite um trabalho futuro estudando novas formas de assinatura, especificamente, um método de assinatura que permita assinar toda um família de documentos, e não apenas um documento.

Pela análise dos métodos de assinatura, podemos dividir a assinatura em duas fases: a fase de resumo e a fase de cifragem. Fica claro que o problema de assinar uma família de documentos é equivalente a encontrar uma função de resumo válida para toda essa família.

7. BIBLIOGRAFIA

1. Alencar Filho, Edgar. Teoria Elementar do Números. Livraria Nobel, 1981.
2. Anderson, John C. Response to NIST's Proposal.Commun. of ACM 35(7), 1992, 50-52.
3. Bellare, Mihir e Micali, Silvio. How to Sign Given Any Trapdoor Permutation, Journal of the Association for Computing Machinery 39(1), jan. 1992, 214-233.
4. Chaum, David. Achieving Electronic Privacy. Scientific American 267(2). Aug. 1992, 96-101.
5. Churchhouse, R.F., Ciphering Algorithms, 1989 CERN School of Computing, 1989, editada po C. Verkerk, 285-308.
6. Denning, D. E. R. Cryptography and Data Security, Addison-Wesley, 1983
7. Diffie, Whitfield e Hellman, Martin E. New Direction in Cryptography. IEEE Trans. on Information Theory IT-22(6) . Nov. 1976, 644-654.
8. El Gamal, T. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory TI-31 (1985) 469-472.
9. Fahn, Paul. Answers to Frequently Asked Questions about Today's Cryptography, RSA Laboratories, 1992.
10. Grollmann, J. and Selman, A. Complexity Measures for Public-Key Cryptosystems. SIAM J. Comput. 17(2). (abril 1988) 309-335.
11. Hellman, Martin E. Response to the NIST's Proposal, Commun. of ACM 35(7), 1992, 47-49.
12. Hellman, Martin E. The Mathematics of Public-Key Cryptography. Scientific American - Trends in Computing, Special Issu/VOl. 1 78-87. Reprinted from Scientific American Aug 1979.
13. Kaliski Jr, Burton S. Some Examples of the PKCS Standards. RSA Data Security Inc., June 1991.
14. Kaliski Jr., Burton S. An Overview of the PKCS Standards. RSA Data Security Inc., June 1991.
15. Kaliski, B. The MD-2 Message-Digest Algorithm, Internet RFC 1319, April 1992.
16. Lempel, A. Cryptology in Transition. ACM Computing Surveys 11(4) , dez. 1979 285-305.
17. Linn, J. Privacy Enhanced for Internet Eletronic Mail: Part I: Message Encription and Authentication Procedures, Internet RFC 1113+, 1992.
18. Lucchesi, C. L. Introdução a Criptografia Computacional. I EBAI, Editora da Unicamp, Campinas, 1985.
19. Lyons, J. W. Summary of the Statement before teh Judiciary Committee of the House of Representatives. Commun. of ACM 35(7), 1992, 53-54..
20. NIST, The Digital Signature Standard. Commun. of ACM 35(7), 1992, 36-40.
21. Popek, G. J. e Kline, C. S. Encryption and Secure Computer Networks. ACM Computing Surveys 11(4) , dez. 1979 331-356.
22. Rivest, R. The MD4 Message-Digest Algorithm, Internet RFC 1320, April 1992.
23. Rivest, R. The MD5 Message-Digest Algorithm, Internet RFC 1321, April 1992.

24. Rivest, R.L., Shamir, A., Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Commun. of ACM 21. 1978, 120-126.

25. Rivest, Ronald L. Response to the NIST's Proposal, Commun. of ACM 35(7), 1992, 41-47

26. RSA Data Security, Inc. PKCS #1 RSA Encription Standard, version 1.4, June 1991.

27. RSA Data Security, Inc. PKCS #3 Diffie-Hellman Key-Agreement Standard, version 1.3, June 1991.

28. RSA Data Security, Inc. PKCS #5 Password-Based Encryption Standard, version 1.4, June 1991.

29. RSA Data Security, Inc. PKCS #6 Extended-Certificate Syntax Standard, version 1.4, June 1991.

30. RSA Data Security, Inc. PKCS #7 Cryptographic Message Syntax Standard, version 1.4, June 1991.

31. RSA Data Security, Inc. PKCS #8 Private-Key Information Syntax Standard, version 1.1, June 1991.

32. RSA Data Security, Inc. PKCS #9 Selected Attribute Types, version 1.0, June 1991.

33. Schnorr, C. P. Efficient identification and signatures for smart cards. Advances in Cryptology: Proceedings of Crypto 89. G. Brassard Ed., Lecutre Notes in Computer Science 435, Springer Verlag, N.Y., 239-251.

34. Simmons, G. J. Symmetric and Asymmetric Encryption. ACM Computing Surveys 11(4) , dez. 1979 305-330.

35. Williams, H. C. A Modification of the RSA Public-Key Encryption Procedure. IEEE Transactions on Information Theory. IT-26(6) Nov. 1980.

36. Woo, Thomas Y.C. e Lam, Simon S. Authentication for Distributed Systems. IEEE Computer, Jan 92, 39-51.

8. INDEX